Efficient Architecture and Design of an Embedded Video Coding Engine

Jie Chen, *Member, IEEE*, and K. J. Ray Liu, *Senior Member, IEEE*

***Abstract*—By doubling the accuracy of motion compensation from integer-pel to half-pel, we can significantly improve the coding gain. Therefore, in this paper, we propose a novel COordinate Rotation Digital Computer (CORDIC) architecture for combined design of discrete cosine transform (DCT) and half-pel motion estimation. Unlike the conventional block matching approaches based on interpolated images, our CORDIC design can directly extracts motion vectors at half-pel accuracy in the transform domain without interpolation. Compared to the conventional block matching methods with interpolation, our multiplier-free design achieves significant hardware saving and far less data flow. Our emphasis in this paper is on achieving efficient design of video coding engine by minimizing computational units along the data path. Furthermore, we implement the embedded design on a dedicated single chip to demonstrate its performance. The DCT-based nature of our design enables us to efficiently combine both DCT and motion estimation units, which are the two most important components of many multimedia standards consuming more than 80% of computing power for a video coder, into one single component. As a result, we can provide a single chip solution for video coding engine while many conventional designs may require multiple chips. In addition, all multiply-and-add (MAC) operations in plane rotations are replaced by CORDIC processors with simple shift-and-add, which is quite simple and compact to realize while being no slower than the bit serial multipliers widely proposed for VLSI array structures. Based on the test result, our chip can operate at 20 MHz with 0.8-μm CMOS technology. Overall, we provide a low-complexity, high throughput solution in this paper for MPEG-1, MPEG-2, and H.263 compatible video codec design.**

***Index Terms*—CORDIC, half-pel motion estimation, video coding, VLSI implementation.**

I. Introduction

In recent years, the technical development in the area of audio-visual communications, notably in video coding, encouraged the emergence of multimedia services which are already changing our everyday life ranging from the low bit-rate, high compression rate video phone applications to the high bit-rate, high-quality high-definition television (HDTV). The motion-compensated discrete cosine transform (DCT) video compression scheme serves as the basis of the multimedia standards such as H.263, MPEG-1, MPEG-2, and MPEG-4. In addition, the block matching scheme for motion estimation and compensation, and the DCT for spatial transformation are the basic techniques in those standards [1]. Efficient coding is accomplished by adding the quantization and entropy coding steps after the DCT block, as shown in Fig. 1. Among those components in Fig. 1, both DCT and motion estimation are computationally intensive units and serve as the computing engine in the video coding systems. One of the virtues of the video coding standards are that those standards only define the video coding syntax and the decoder structures but leave the room for creativity of the encoder design. In other words, the encoders is MPEG compatible as long as they can produce the standard video coding syntax. Therefore, in this paper, we propose a novel CORDIC architecture of an embedded video coding engine for MPEG compatible video coding system. We focus on the combined design and implementation of DCT and half-pel motion estimation because

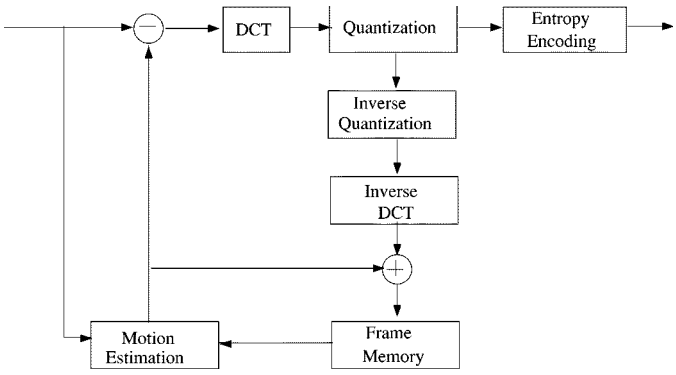

Fig. 1. Conventional hybrid motion-compensated video coder structure. (Here, motion estimation is achieved in spatial domain.)

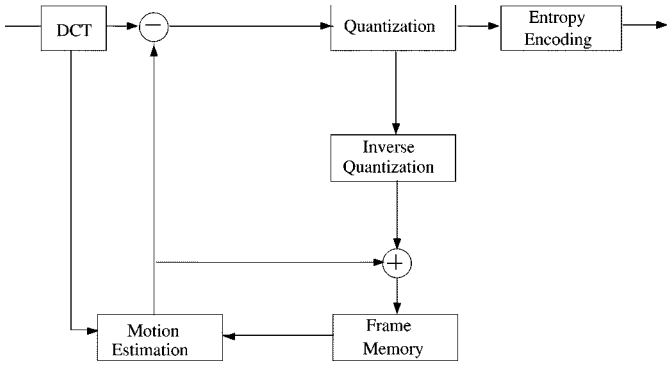

Fig. 2. Fully DCT-based video coder structure. (Here, motion estimation is achieved in DCT domain.)

Manuscript received October 21, 1999; revised March 6, 2001. This work was supported in part by the ONR Grant N00014-93-10566 and the NSF NYI Award MIP9457397. The associate editor coordinating the review of this paper and approving it for publication was Prof. Chung-Yu Wu.

J. Chen was with Bell Laboratories, Lucent Technologies, Murray Hill, NJ 07974 USA. He is now with Flarion Technologies, Bedminster, NJ 07921USA (e-mail: chenjie@Glue.umd.edu).

K. J. R. Liu is with the Electrical Engineering Department and Institute for Systems Research, University of Maryland, College Park, MD 20742 USA (e-mail: kjrliu@eng.umd.edu).

Publisher Item Identifier S 1520-9210(01)07785-9.

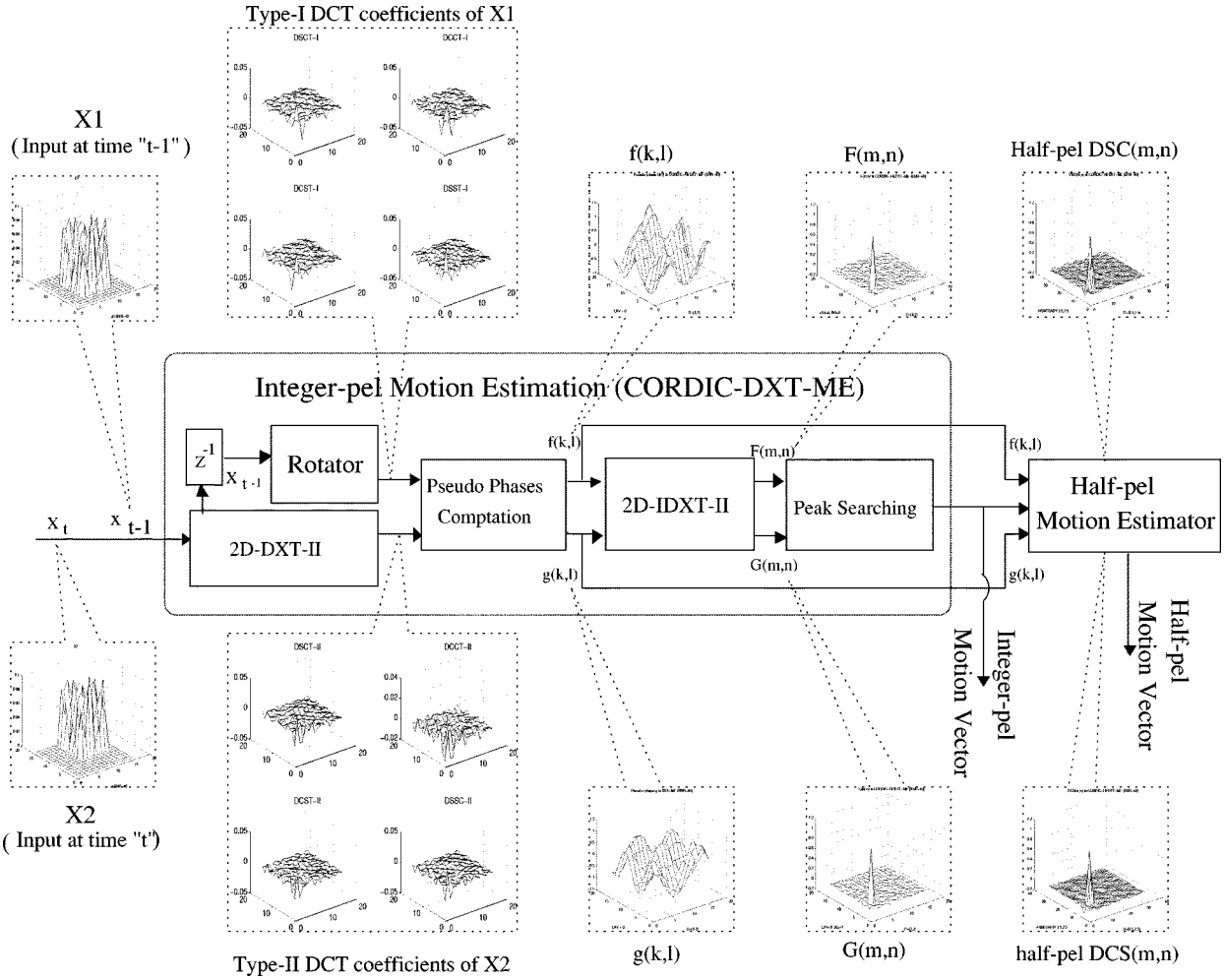

Fig. 3. Block diagram of the embedded video coding engine. Here, we feed synthetic data into the design to envision how it works. The waveforms along the data path correspond to the outputs at those points.

they are the core elements consuming more than 80% of computing power [2]–[4].

The objective of block-based motion estimation is to determine the blocks in the reference frame that better matches (according to mean square error or mean absolute difference criterion) the characteristics of the blocks in the current frame. The most commonly used motion estimation architectures are based on the block matching motion estimation (BKM-ME) [5]–[9]. Due to the complexity of block matching, the hardware complexity is also high. To reduce the hardware complexity, architectures for block matching based on hierarchical search strategies were presented in [10] and [11]. These structures require two or more sequential steps/stages to find suboptimal estimates. However, for those block-based approaches, there is a well-known side effect—"blocking effect." To overcome this effect, lapped orthogonal transform (LOT)- [12] and complex lapped transform (CLT)-based motion estimation were therefore proposed [13]. The shortcoming is that they still require searching over a larger area and thus result in a very high computational burden. In addition, motion estimation using CLT-ME is accurate on moving sharp edges but not on blur edges. Besides the block-based approaches, pel-based estimation methods such as pel-recursive algorithm (PRA-ME) [14] and architecture [15], optical flow approach [16] have also been proposed. However, the pel-based estimation methods are very vulnerable to noise by virtue of their involving only local operations and may suffer from the instability problem. Other than the block-based approaches used in all video coding standards, the manipulation of video data in the DCT domain has long been recognized important in many advanced video applications [17]. Many manipulation functions can be performed in the DCT domain more efficiently than in the spatial domain. In the category of transform-domain motion estimation, three-dimensional fast Fourier transform (3D-FFT) has been successfully used to estimate motion in several consecutive frames [18]. However, this approach is not compatible with the standards because it requires processing of several frames rather than two. Moreover, the FFT operating on complex numbers is not used in the video standards, and the global routing structure is undesirable in VLSI design. Recently, a DCT-domain motion estimation and compensation design was proposed in [19] to reduce the amount of storage space of an entire image for predictive coding.

To further improve the compression rate and resolution, motion estimation with sub-pixel accuracy is essential because movements in a video sequence are not necessarily multiples of the sampling grid distance. The temporal prediction error

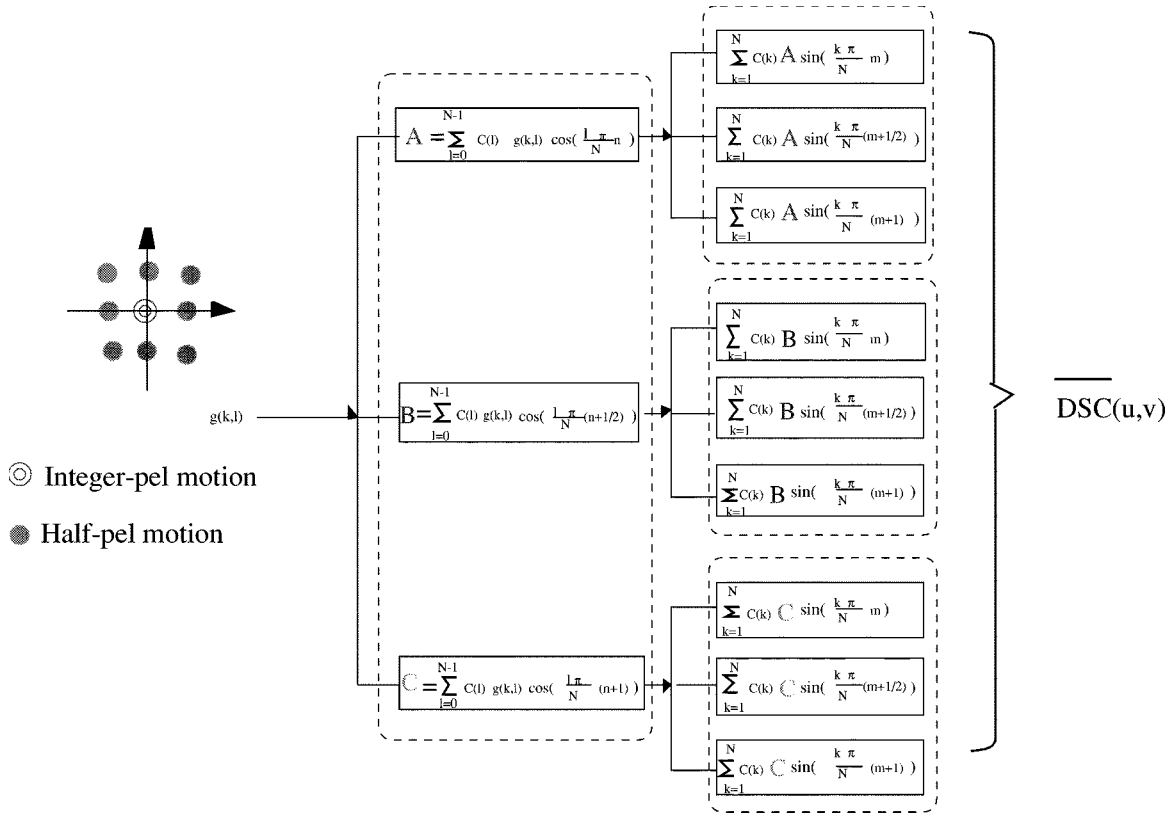

Fig. 4. Schematic diagram of decomposing $\overline{DSC}(u, v)$ computation.

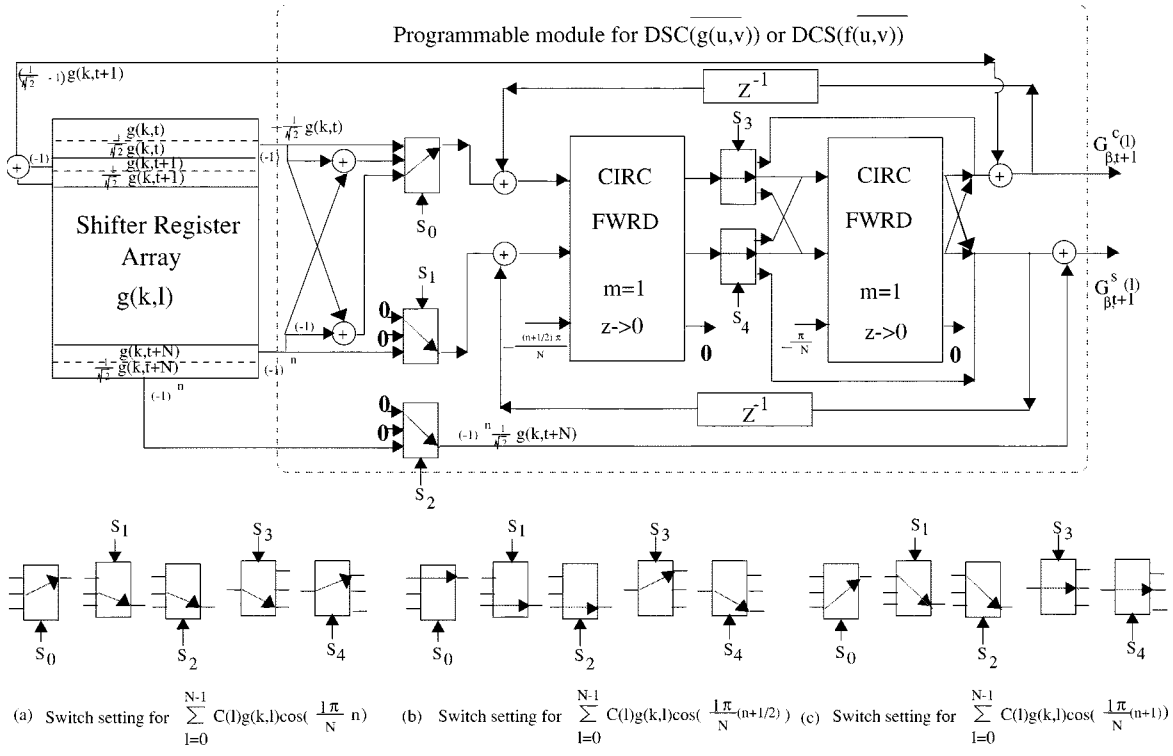

Fig. 5. Programmable unified module to compute $\overline{DSC}(u, v)$.

variance is generally decreased by employing sub-pixel motion compensation. It is shown that doubling the accuracy of motion compensation from integer-pel to half-pel can reduce the bit-rate by up to 0.5 bits/sample [20], [21]. Studies have also revealed that, beyond a certain "critical accuracy," the possibility of further improving prediction by using more accurate motion compensation schemes is small [22] while the hardware complexity increases prohibitively high. As a result,

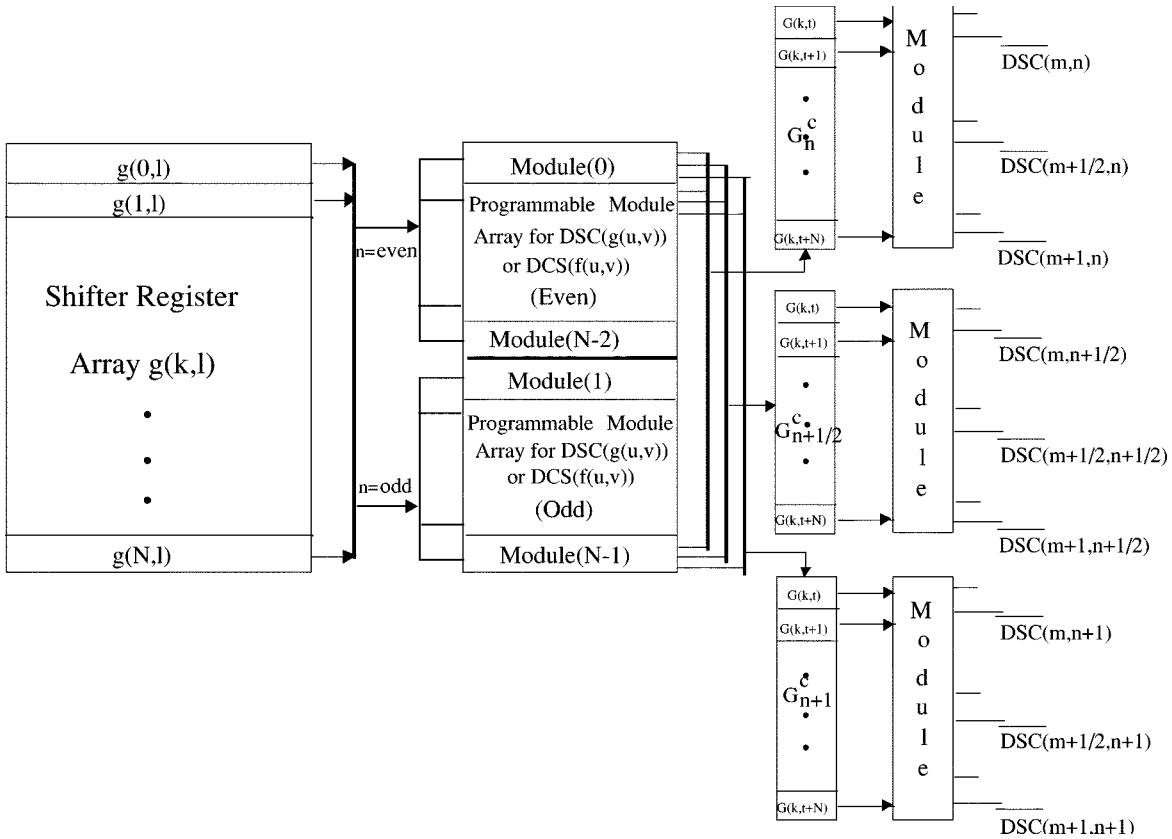

Fig. 6. Block diagram of computing $\overline{DSC}(u, v)$ in parallel.

TABLE I
PARAMETERS USED IN THE UNIFIED EQUATION (18)

Phase	$n+\frac{1}{2}$	n	$n+1$
Δ	$\mathbf{R}(n+\frac{1}{2})$	$\mathbf{R}(n+\frac{1}{2})\cdot\mathbf{R}(-\frac{1}{2})$	$\mathbf{R}(n+\frac{1}{2})\cdot\mathbf{R}(\frac{1}{2})$
Γ	$\begin{bmatrix} A \\ B \end{bmatrix}$	$\begin{bmatrix} A+B \\ 0 \end{bmatrix}$	$\begin{bmatrix} A-B \\ 0 \end{bmatrix}$
α	$(-1)^n\frac{1}{\sqrt{2}}g(k,t+N)$	0	0

motion compensation with half-pel accuracy is recommended in MPEG-1, MPEG-2, MPEG-4, and H.263 standards. Many sub-pixel motion estimation schemes have been proposed [23]–[25]. The most commonly used spatial-domain fractional-pel motion estimation algorithms such as block matching approach [26]–[28] and the pel-recursive approach [29], [30] require interpolation of images through bilinear, Lagrange, or other interpolation methods [31]. In [32], the exhaustive block matching implemented with a one-dimensional (1-D) systolic array was proposed. A half-pel precision processing unit is then introduced to estimate half-pel motion vectors based on those at integer-pel accuracy using bilinear interpolation method. However, interpolation not only increases the complexity and data flow of a coder but also may adversely affect the accuracy of motion estimates from the interpolated images. It is more desirable that sub-pixel accuracy of motion estimates can be obtained without interpolating the images.

In many video coding approaches, motion estimation is performed in the spatial domain while DCT is done in the transform domain. Thus, it results in the video coding system, as shown in Fig. 1. However, this system is not an efficient design because both motion estimation and DCT/IDCT units can not be combined together into one unit. Moreover, the throughput of the coder is limited by the processing speed of major components (DCT, IDCT, spatial domain motion estimation) in the feedback loop, which has been recognized as the bottleneck of video coding systems. In this paper, we emphasis on minimizing the computational units along the data path based on the DCT pseudo-phase technique [33], [34], [45], [46] proposed for motion estimate and compensate in DCT domain at integer and half-pixel accuracy. With such a design, we can move the DCT unit out of the feedback loop and get rid of the inverse discrete cosine transform (IDCT) to achieve a fully DCT-based video coder design, as in Fig. 2. The resulted design not only reduces the system complexity of the coder but also achieves higher data throughput. Due to the DCT-based nature of our design, we can replace all the multiply-and-add (MAC) operations by CORDIC processors which is extremely simple and quite compact to realize while being no slower than the bit serial multipliers widely proposed for VLSI array structures.

03–In this paper, we propose an efficient fully pipelined parallel CORDIC architecture for combined design of DCT and half-pel motion estimation, which serves as an embedded video coding engine. In addition, we implement it on a dedicated single chip to demonstrate its performance. In what follows, we discuss the parallel architecture for half-pel motion estimation and present the simulation results of our design. In Section III, we implement the design on a dedicated single chip to demonstrate its performance. Finally, the paper is concluded in Section IV.

(a) (b) (c)

Fig. 7. Subjective test of our design using slow motion video sequence "News" (a) previous frame; (b) current frame; and (c) reconstructed current frame based on CORDIC-HDXT-ME outputs.

(a) (b) (c)

Fig. 8. Subjective test of our design using medium motion video sequence "table tennis" (a) previous frame; (b) current frame; and (c) reconstructed current frame based on CORDIC-HDXT-ME outputs.

II. CORDIC Architecture for Video Coding Engine

As we have mentioned in Section I, it is more desirable to estimate motion at half-pel accuracy *without* any inter-pixel interpolation used in the video coding standards so that seamless integration of the motion compensation unit with the spatial compression unit is possible. The resulted design can be more efficient by performing the video coding solely in the DCT domain. Based on the concept of pseudo-phase technique, we propose the fully pipelined parallel CORDIC-based processing unit which not only computes the DCT coefficients but also estimates motion vectors at half-pel accuracy. The overall architecture (CORDIC-HDXT-ME) is outlined in Fig. 3. The design is fully pipelined because we estimation the motion at integer-pel accuracy based on the corresponding DCT coefficients. In addition, the half-pel motion vectors are computed based on motion vectors at integer-pel accuracy encircled by the box in Fig. 3. Within each computational block, we process the data in the parallel fashion. In what follows, we will describe the detail design of each block in Fig. 3.

A. DCT and Half-Pel Motion Estimation

The input image is divided into nonoverlapped blocks of $N \times N$ pixels. Here, the block size N is adjustable and the motion vectors are computed per block basis. Notice that the motion vectors are limited to the block size. If the motion vectors go beyond the block boundary, the motion vector of $(0, 0)$ will be used instead. Our embedded video coding engine serially takes the block of pixels staring from the top left pixel to the top right one, and then on to the next row as input. The process continues until the entire input image is estimated. The DCT and half-pel motion estimation consists of five major processing stages/steps:

1) The $N \times N$ block of pixels at the current frame x_t are serially fed into "*2-D-DXT-II*" block—the type-II [35], [36] 2-D DCT and discrete sine transform (DST) coder. It computes four DXT(DCT/DST) coefficients:$X_t^{cc}(k, l)$, $X_t^{cs}(k, l)$, $X_t^{sc}(k, l)$, and $X_t^{ss}(k, l)$ simultaneously such as

$$X_t^{sc}(k, l) = \frac{4}{N^2} C(k)C(l) \sum_{m,n=0}^{N-1} x_t(m, n) \cdot \sin\left[\frac{k\pi}{N}(m+0.5)\right] \cos\left[\frac{l\pi}{N}(n+0.5)\right] \quad \text{for} \quad k \in \{1, \ldots, N\}, l \in \{0, \ldots, N-1\} \tag{1}$$

where $X_t^{cc}(k, l)$ is the DCT used in video standards to reduce the spatial domain redundancy. Meanwhile the previously calculated four DXT coefficients of the previous frame x_{t-1} (delayed by one clock cycle as denoted by "Z^{-1}" in Fig. 3) are now transformed to type-I 2-D DCT/DST coefficients $Z_{t-1}^{cc}(k, l)$, $Z_{t-1}^{cs}(k, l)$, $Z_{t-1}^{sc}(k, l)$ and $Z_{t-1}^{ss}(k, l)$ simultaneously such as

$$Z_{t-1}^{cs}(k, l) = \frac{4}{N^2} C(k)C(l) \sum_{m=0}^{N-1} \sum_{n=0}^{N-1} x_{t-1}(m, n) \cdot \cos\left[\frac{k\pi}{N}(m)\right] \sin\left[\frac{l\pi}{N}(n)\right] \quad k \in \{0, \ldots, N\}, l \in \{1, \ldots, N-1\}. \tag{2}$$

The reason we use "*Rotator*" block in Fig. 3 is because those type-II to type-I DXT coefficients conversion can be realized by simple plane rotation

$$\begin{bmatrix} \cos\theta\cos\phi & \cos\theta\sin\phi & \sin\theta\cos\phi & \sin\theta\sin\phi \\ -\cos\theta\sin\phi & \cos\theta\cos\phi & -\sin\theta\sin\phi & \sin\theta\cos\phi \\ -\sin\theta\cos\phi & -\sin\theta\sin\phi & \cos\theta\cos\phi & \cos\theta\sin\phi \\ \sin\theta\sin\phi & -\sin\theta\cos\phi & -\cos\theta\sin\phi & \cos\theta\cos\phi \end{bmatrix}$$

$$\begin{bmatrix} X_{t-1}^{cc}(k,l) \\ X_{t-1}^{cs}(k,l) \\ X_{t-1}^{sc}(k,l) \\ X_{t-1}^{ss}(k,l) \end{bmatrix} = \begin{bmatrix} Z_{t-1}^{cc}(k,l) \\ Z_{t-1}^{cs}(k,l) \\ Z_{t-1}^{sc}(k,l) \\ Z_{t-1}^{ss}(k,l) \end{bmatrix} \tag{3}$$

where $\theta = (k\pi/2N)$, $\phi = (l\pi/2N)$.

2) The "*Pseudo-Phases Computation*" module utilizes all the previous calculated type-I/type-II DXT parameters and takes $O(N)$ times to produce two pseudo-phase functions $f(k, l)$ and $g(k, l)$

$$\underbrace{\begin{bmatrix} Z_{t-1}^{cc}(k,l) & -Z_{t-1}^{cs}(k,l) & -Z_{t-1}^{sc}(k,l) & Z_{t-1}^{ss}(k,l) \\ Z_{t-1}^{cs}(k,l) & Z_{t-1}^{cc}(k,l) & -Z_{t-1}^{ss}(k,l) & -Z_{t-1}^{sc}(k,l) \\ Z_{t-1}^{sc}(k,l) & -Z_{t-1}^{ss}(k,l) & Z_{t-1}^{cc}(k,l) & -Z_{t-1}^{cs}(k,l) \\ Z_{t-1}^{ss}(k,l) & Z_{t-1}^{sc}(k,l) & Z_{t-1}^{cs}(k,l) & Z_{t-1}^{cc}(k,l) \end{bmatrix}}_{\mathbf{Z}_{t-1}(k,l)}$$

$$\underbrace{\begin{bmatrix} \star \\ f(k,l) \\ g(k,l) \\ \star \end{bmatrix}}_{\vec{\theta}_{m,n}(k,l)} = \underbrace{\begin{bmatrix} X_t^{cc}(k,l) \\ X_t^{cs}(k,l) \\ X_t^{sc}(k,l) \\ X_t^{ss}(k,l) \end{bmatrix}}_{\vec{x}_t(k,l)}$$

$$\text{for} \quad k, l \in \{1, \ldots, N-1\} \tag{4}$$

where $\vec{\theta}_{m,n}$ is the pseudo-phase vector and $\star$ stands for "don't care."

3) Then $f(k, l)$ and $g(k, l)$ undergo "*2-D-IDXT-II*" block—the type-II 2-D inverse DCT and DST (IDCT/IDST) transform to generate $F(m, n)$ and $G(m, n)$ in view of the orthogonal property

$$\begin{aligned} F(m,n) &= IDCSTII(f(k,l)) \\ &= \frac{4}{N^2}\sum_{k=0}^{N-1}\sum_{l=1}^{N} C(k)C(l)f(k,l) \\ &\quad \cdot \cos\frac{k\pi}{N}\left(m+\frac{1}{2}\right)\sin\frac{l\pi}{N}\left(n+\frac{1}{2}\right) \\ &= [\delta(m-m_u)+\delta(m+m_u+1)] \\ &\quad \cdot [\delta(n-m_v)-\delta(n+m_v+1)] \end{aligned} \tag{5}$$

$$\begin{aligned} G(m,n) &= IDSCTII(g(k,l)) \\ &= \frac{4}{N^2}\sum_{k=1}^{N}\sum_{l=0}^{N-1} C(k)C(l)g(k,l) \\ &\quad \cdot \sin\frac{k\pi}{N}\left(m+\frac{1}{2}\right)\cos\frac{l\pi}{N}\left(n+\frac{1}{2}\right) \\ &= [\delta(m-m_u)-\delta(m+m_u+1)] \\ &\quad \cdot [\delta(n-m_v)+\delta(n+m_v+1)] \end{aligned} \tag{6}$$

where m_v, m_u are the peak positions used in determining the integer-pel motion vectors.

4) We search the peak values among $F(m, n)$ and $G(m, n)$ to determine the integer-pel motion vector (m_u, m_v). The sign of the peak values indicates the direction of the movement i.e., (-,-) stands for southwest. If the motion goes beyond the block boundary, motion vector of $(0, 0)$ will be used, instead. Therefore, it depends very much on the block size whether we can find the motion vectors or not for those big motion video sequences (the same situation can be encountered for the block matching scheme used in those video standards). In our design, the block size is adjustable based upon the characters of the input video sequences. Notice that the peak search used here is totally different and much simpler than the block matching search used in the standards because we only look for the maximum value among two-dimensional (2-D) data array.

5) Unlike the conventional half-pel motion estimation design based on the interpolated image, our design can extract the motion at half-pel accuracy without any interpolation. In principle, we compute the motion vectors at half-pel accuracy based upon the integer-pel motion vector (m, n) ($m = m_u$ and $n = m_v$, which is obtained previously). We utilize the "*half-pel motion estimator*" in Fig. 3 block to produce $\overline{DCS}(u, v)$ and $\overline{DSC}(u, v)$ for $u \in \{m-0.5, m, m+0.5\}$ and $v \in \{n-0.5, n, n+0.5\}$. Both $\overline{DCS}(u, v)$ and $\overline{DSC}(u, v)$ are defined as follows: [34], [46]:

$$\begin{aligned} \overline{DCS}(u,v) &= \sum_{k=0}^{N-1}\sum_{l=1}^{N} C(k)C(l)f(k,l) \\ &\quad \cdot \cos\frac{k\pi}{N}\left(u+\frac{1}{2}\right)\sin\frac{l\pi}{N}\left(v+\frac{1}{2}\right) \\ &= \sum_{k=0}^{N-1}\sum_{l=1}^{N} C(k)C(l)f(k,l) \\ &\quad \cdot \cos\frac{k\pi}{N}\left(m+\mu_u+\frac{1}{2}\right)\sin\frac{l\pi}{N}\left(n+\mu_v+\frac{1}{2}\right) \end{aligned} \tag{7}$$

$$\begin{aligned} \overline{DSC}(u,v) &= \sum_{k=1}^{N}\sum_{l=0}^{N-1} C(k)C(l)g(k,l) \\ &\quad \cdot \sin\frac{k\pi}{N}\left(u+\frac{1}{2}\right)\cos\frac{l\pi}{N}\left(v+\frac{1}{2}\right) \\ &= \sum_{k=1}^{N}\sum_{l=0}^{N-1} C(k)C(l)g(k,l) \\ &\quad \cdot \sin\frac{k\pi}{N}\left(m+\mu_u+\frac{1}{2}\right)\cos\frac{l\pi}{N}\left(n+\mu_v+\frac{1}{2}\right) \end{aligned} \tag{8}$$

where μ_u, $\mu_v \in \{-0.5, 0, 0.5\}$. Therefore, the half-pel motion vector is determined by only considering the nine possible positions $(m+\mu_u, n+\mu_v)$ around the integer-pel

displacement (m, n). Similar to the integer-pel design, the peak position of either $\overline{DCS}(u, v)$ or $\overline{DSC}(u, v)$ determines the half-pel motion vector.

To envision how our embedded video coding engine works, we feed the synthetic data into our design, as shown in Fig. 3. The waveforms along the data path correspond to the outputs at those points. By moving the block of pixels, $X1$, of synthetic data in the direction of (3.5, 2.5), we get $X2$, as shown in Fig. 3 with the additive Gaussian noise of signal-to-noise (SNR) $= 40$ dB. We feed $X1$ at time "$t-1$" (the previous frame) followed by $X2$ at time "t" (the current frame). After the first stage of our CORDIC design, the *type-II* 2-D DCT/DST coefficients of the current frame ($X_t^{cc}(k, l)$, $X_t^{cs}(k, l)$, $X_t^{sc}(k, l)$, and $X_t^{ss}(k, l)$) and the *type-I* 2-D DCT/DST coefficients of the previous frame ($Z_{t-1}^{cc}(k, l)$, $Z_{t-1}^{cs}(k, l)$, $Z_{t-1}^{sc}(k, l)$, and $Z_{t-1}^{ss}(k, l)$) are generated simultaneously. Here, the *type-I* coefficients are generated by simply rotating those *type-II* DXT coefficients stored in the register array and delayed by one clock cycle as denoted by Z^{-1}. After the second stage, both pseudo-phase functions—$f(k, l)$ and $g(k, l)$ are computed, as shown in Fig. 3. Then $f(k, l)$ and $g(k, l)$ functions undergo the *2-D-IDXT-II* transform at the third stage to generate the corresponding $F(m, n)$ and $G(m, n)$, respectively. The peak position among $F(m, n)$ and $G(m, n)$ implies the integer-pel motion vector of $(3, 2)$. In addition, the signs of peak value $(+, +)$ reveal the direction of the movement—toward northeast. Based on the previously calculated results, both $\overline{DSC}(u, v)$ and $\overline{DCS}(u, v)$ are computed by the "*half-pel motion estimator*." The peaks among those computed value indicate the corresponding half-pel motion of $(3.5, 2.5)$ as we expected.

From the above discussion, we observe that the half-pel motion estimation unit works solely in DCT domain without interpolation of input images. Therefore, it achieves significant savings in hardware complexity and far less data flow compared to the conventional half-pel block matching approaches based on interpolated images. Due to the DCT-based nature of our design, the inherent advantages are summarized as follows:

- **Low complexity**: Compared to $O(S^2 \cdot N^2)$ complexity for the commonly used half-pel Block Matching design [32], we can compute both DCT and half-pel motion estimation at lower $O(S^2)$ complexity. Here, S stands for the search range and N stands for the block size. Therefore, it makes our design attractive in real-time multimedia applications. For instance, the total number of operations in our design are 7168 with the block size of 16×16 and 28 672 with the block size of 32×32 while the block matching scheme requires 130 816 operations with block size of 16×16 and search area size of 32×32 (please refer to our discussion later).
- **No multiplication**: Because of the DCT-based design, we can replace all MAC operations and trigonometric calculations in plane rotation by CORDIC processors.
- **Suitable for VLSI implementation**: The regular, modular and only locally connected properties of our embedded video coding engine are very suitable for VLSI implementation.

In what follows, we will discuss the detail design about solving $\overline{DCS}(u, v)$ and $\overline{DSC}(u, v)$ for half-pel motion estimation (please refer to [37], [47] for the combined design of DCT and integer-pel motion estimation).

B. Half-Pel Motion Estimator Design

By taking a close look at (7) and (8), we observe that both $\overline{DCS}(u, v)$ and $\overline{DSC}(u, v)$ computations are similar. Thus we will use $\overline{DSC}(u, v)$ computation as an example to explain our design, the same approach can be applied to calculate $\overline{DCS}(u, v)$. In order to figure out 2-D $\overline{DSC}(u, v)$, we can decompose the computation into cascade 1-D *type-II* inverse DCT calculations, as illustrated in Fig. 4. Actually those computations encircled by dot-boxes, such as

$$\begin{aligned} A &= \sum_{l=0}^{N-1} C(l) g(k, l) \cos\left(\frac{l\pi}{N} n\right); \\ B &= \sum_{l=0}^{N-1} C(l) g(k, l) \cos\left(\frac{l\pi}{N}\left(n+\frac{1}{2}\right)\right) \\ C &= \sum_{l=0}^{N-1} C(l) g(k, l) \cos\left(\frac{l\pi}{N}(n+1)\right) \end{aligned} \tag{9}$$

in the middle level of Fig. 4, can be grouped together and replaced by a unified programmable module, as shown in Fig. 5. The reason is that those computations are the same except for the phase differences such as $(l\pi/N)n$, $(l\pi/N)(n+(1/2))$ and $(l\pi/N)(n+1)$. Next, we will explain how to combine those computations together and come out a programmable unified design.

Let us define

$$G_{\beta,t}^{c}(l) = \frac{2}{N} \sum_{l=t}^{t+N-1} C(l-t) g(k, l) \cos\left[\frac{(l-t)\pi}{N}(n+\beta)\right]$$

$$G_{\beta,t}^{s}(l) = \frac{2}{N} \sum_{l=t+1}^{t+N} C(l-t) g(k, l) \sin\left[\frac{(l-t)\pi}{N}(n+\beta)\right]$$

where

$$C(l) = \begin{cases} \dfrac{1}{\sqrt{2}}, & \text{for } l = 0 \text{ or } N \\ 1, & \text{otherwise,} \end{cases} \tag{10}$$

with $\beta \in \{0, 1/2, 1\}$. The time index t in $G_{\beta,t}^{c}(l)$ and $G_{\beta,t}^{s}(l)$ denotes that the transform starts from $g(k, t)$. And an auxiliary variable

$$G_{\beta,t}^{as}(l) = \frac{2}{N} \sum_{l=t}^{t+N-1} C(l-t) g(k, l) \sin\left[\frac{(l-t)\pi}{N}(n+\beta)\right] \tag{11}$$

is introduced to maintain the lattice structure so that we can replace all the trigonometric in plane rotation by CORDIC processor. Notice that $G_{\beta,t}^{s}(l)$ is obtained indirectly through $G_{\beta,t}^{as}(l)$.

For phase $\dfrac{l\pi}{N}\left(n+\dfrac{1}{2}\right)$.

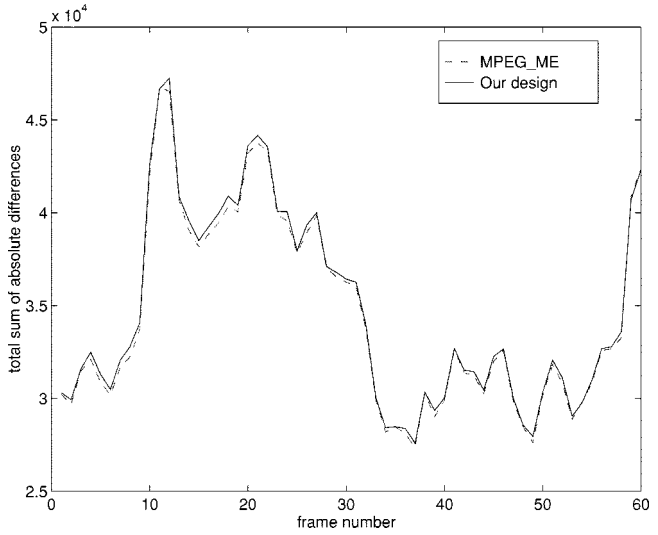

Fig. 9. Comparison of prediction errors between our design and the conventional block matching motion estimation (MPEG-ME) using "News" test sequence. Here, the total sum of absolute differences is the absolute difference computed on per frame basis.

Based on the definitions in (10) and (11), we can apply time-recursive approach to solve for $G^c_{1/2,\,t+1}(l)$ and $G^{as}_{1/2,\,t+1}(l)$ as follows:

$$\begin{bmatrix} G^c_{1/2,\,t+1}(l) \\ G^{as}_{1/2,\,t+1}(l) \end{bmatrix} = \begin{bmatrix} \cos\frac{\left(n+\frac{1}{2}\right)\pi}{N} & \sin\frac{\left(n+\frac{1}{2}\right)\pi}{N} \\ -\sin\frac{\left(n+\frac{1}{2}\right)\pi}{N} & \cos\frac{\left(n+\frac{1}{2}\right)\pi}{N} \end{bmatrix} \cdot \begin{bmatrix} \overline{G^c}_{1/2,\,t+1}(l) \\ \overline{G^{as}}_{1/2,\,t+1}(l) \end{bmatrix} \tag{12}$$

where $\overline{G^c}_{1/2,\,t+1}(l)$ is related to $G^c_{1/2,\,t}(l)$ and $\overline{G^{as}}_{1/2,\,t+1}(l)$ is related to $G^{as}_{1/2,\,t}(l)$ by:

$$\begin{aligned} &\overline{G^c}_{1/2,\,t+1}(l) \\ &= \frac{2}{N}\sum_{l=t+1}^{t+N} C(l-t-1)g(k,\,l)\cos\left[\frac{(l-t)\pi}{N}\left(n+\frac{1}{2}\right)\right] \\ &= \frac{2}{N}\sum_{l=t+2}^{t+N-1} g(k,\,l)\cos\left[\frac{(l-t)\pi}{N}\left(n+\frac{1}{2}\right)\right] \\ &\quad + \frac{2}{N}\frac{1}{\sqrt{2}}g(k,\,t+1)\cos\left[\frac{\pi}{N}\left(n+\frac{1}{2}\right)\right] \\ &= G^c_{1/2,\,t}(l) - \frac{2}{N}\frac{1}{\sqrt{2}}g(k,\,t) + \frac{2}{N} \\ &\quad \cdot\left(\frac{1}{\sqrt{2}}-1\right)g(k,\,t+1)\cos\left[\frac{\pi}{N}\left(n+\frac{1}{2}\right)\right] \end{aligned} \tag{13}$$

$$\begin{aligned} &\overline{G^{as}}_{1/2,\,t+1}(l) \\ &= \frac{2}{N}\sum_{l=t+1}^{t+N} C(l-t-1)g(k,\,l)\sin\left[\frac{(l-t)\pi}{N}\left(n+\frac{1}{2}\right)\right] \\ &= \frac{2}{N}\sum_{l=t+2}^{t+N-1} g(k,\,l)\sin\left[\frac{(l-t)\pi}{N}\left(n+\frac{1}{2}\right)\right] \\ &\quad + \frac{2}{N}\frac{1}{\sqrt{2}}g(k,\,t+1)\sin\left[\frac{\pi}{N}\left(n+\frac{1}{2}\right)\right] \\ &\quad + (-1)^n\frac{2}{N}g(k,\,t+N) \\ &= G^{as}_{1/2,\,t}(l) + (-1)^n\frac{2}{N}g(k,\,t+N) + \frac{2}{N} \\ &\quad \cdot\left(\frac{1}{\sqrt{2}}-1\right)g(k,\,t+1)\sin\left[\frac{\pi}{N}\left(n+\frac{1}{2}\right)\right]. \end{aligned} \tag{14}$$

To maintain the lattice structure, the auxiliary variable $G^{as}_{1/2,\,t}(l)$ is related to $G^s_{1/2,\,t}(l)$ by

$$\begin{aligned} G^s_{1/2,\,t}(l) &= \frac{2}{N}\sum_{l=t+1}^{t+N} C(l-t)g(k,\,l)\sin\left[\frac{(l-t)\pi}{N}\left(n+\frac{1}{2}\right)\right] \\ &= \frac{2}{N}\sum_{l=t+1}^{t+N-1} g(k,\,l)\sin\left[\frac{(l-t)\pi}{N}\left(n+\frac{1}{2}\right)\right] \\ &\quad + (-1)^n\frac{2}{N}\frac{1}{\sqrt{2}}g(k,\,t+N) \\ &= G^{as}_{1/2,\,t}(l) + (-1)^n\frac{2}{N}\frac{1}{\sqrt{2}}g(k,\,t+N). \end{aligned} \tag{15}$$

Based on the above derivations, we can combine those equations together and get

$$\begin{cases} \begin{bmatrix} G^c_{1/2,\,t+1}(l) \\ G^{as}_{1/2,\,t+1}(l) \end{bmatrix} = \mathbf{R}\left(n+\frac{1}{2}\right) \\ \quad \cdot\left[\begin{bmatrix} G^c_{1/2,\,t}(l) \\ G^{as}_{1/2,\,t}(l) \end{bmatrix} + \frac{2}{N}\begin{bmatrix} -\frac{1}{\sqrt{2}}g(k,\,t) \\ (-1)^n g(k,\,t+N) \end{bmatrix}\right] \\ \quad + \begin{bmatrix} \frac{2}{N}\left(\frac{1}{\sqrt{2}}-1\right)g(k,\,t+1) \\ 0 \end{bmatrix} \\ G^s_{1/2,\,t}(l) = G^{as}_{1/2,\,t}(l) + (-1)^n\frac{2}{N}\frac{1}{\sqrt{2}}g(k,\,t+N) \end{cases} \tag{16}$$

where

$$\mathbf{R}\left(n+\frac{1}{2}\right) = \begin{bmatrix} \cos\frac{\left(n+\frac{1}{2}\right)\pi}{N} & \sin\frac{\left(n+\frac{1}{2}\right)\pi}{N} \\ -\sin\frac{\left(n+\frac{1}{2}\right)\pi}{N} & \cos\frac{\left(n+\frac{1}{2}\right)\pi}{N} \end{bmatrix}. \tag{17}$$

Unified Design

With the same approach, we can get the time-recursive updated relations to compute $G^c_{\beta,\,t}(l)$, $G^s_{\beta,\,t}(l)$ for different phases

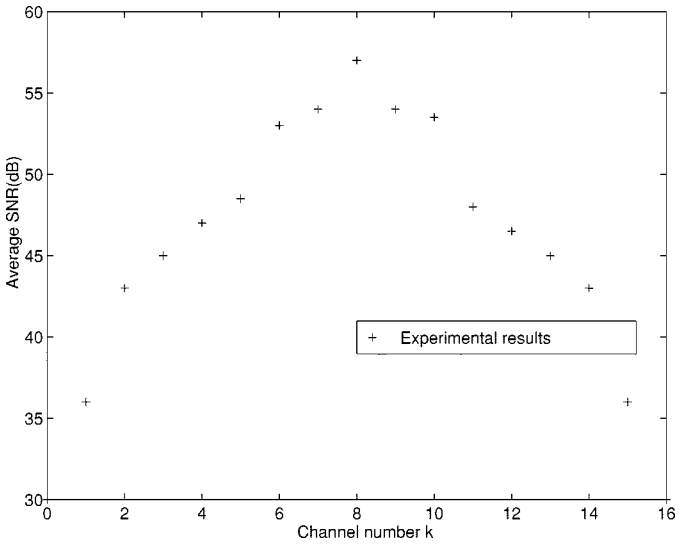

Fig. 10. Average SNR as a function of channel number k in fixed-point analysis with the word-length of $B = 12$.

such as $(l\pi/N)n$ and $(l\pi/N)(n+1)$. The unified equation is given as follows:

$$\begin{cases} \begin{bmatrix} G^{c}_{\beta,\,t+1}(l) \\ G^{as}_{\beta,\,t+1}(l) \end{bmatrix} = \mathbf{\Delta} \left[\begin{bmatrix} G^{c}_{\beta,\,t}(l) \\ G^{as}_{\beta,\,t}(l) \end{bmatrix} + \mathbf{\Gamma} \right] \\ + \begin{bmatrix} \left(\dfrac{1}{\sqrt{2}} - 1 \right) g(k,\, t+1) \\ 0 \end{bmatrix} \\ G^{s}_{\beta,\,t}(l) = G^{as}_{\beta,\,t}(l) + \alpha. \end{cases} \tag{18}$$

where

$$A = -\frac{2}{N}\,\frac{1}{\sqrt{2}}\,g(k,\,t), \qquad B = \frac{2}{N}\,\left(\frac{1}{\sqrt{2}} - 1\right) g(k,\,t{+}1). \tag{19}$$

Also, parameters $\mathbf{\Delta}$, $\mathbf{\Gamma}$, α depending on different phases are listed in Table I. The programmable unified structure to realize (18) is shown in Fig. 5, where the switch settings are controlled by different phases. Each programmable module in Fig. 5 requires two CORDICs.

With different inputs and rotation angles, N such modules, as shown in Fig. 5 can be used to compute $\overline{DSC}(u,\,v)$ in parallel, as shown in Fig. 6. It takes $O(N)$ time to produce $\overline{DSC}(u,\,v)$ for $u \in \{m - 0.5,\, m,\, m + 0.5\}$, $v \in \{n - 0.5,\, n,\, n + 0.5\}$ and $O(1)$ to find the peak position corresponding to the half-pel motion vector. Overall, the half-pel motion estimator needs the total of $2N + 6$ CORDIC processors and $4N + 18$ adders.

C. Simulation Results

We have applied synthetic data with additive noise, slow to medium motion video sequences such as "Claire," "News," and "TableTennis" as the input test sequences of our embedded video coding engine to check its performance. As a subjective measurement of our design (CORDIC-HDXT-ME), the original previous and current frames of "News" and "Table Tennis" sequences are shown in Fig. 7(a) and (b) and Fig. 8(a) and (b), respectively. The reconstructed current frames of "News" and "Table Tennis" sequences based on the motion vectors obtained from our design is shown in Fig. 7(c) and Fig. 8(c). Here, we use the MPEG-2 decoder to reconstruct the video sequences. We observe that the reconstructed frame has compatible video quality as the original one. The simulation results of prediction errors using both our design and the conventional block matching motion estimation are plotted in Fig. 9 using the first 100 frames of "News" video sequence. Again, the simulation shows that the comparable performance of our design and the block matching approaches.

III. VLSI Design for Embedded Video Coding Engine

The regular, modular, and locally connected architecture of our design is suitable for VLSI implementation. In addition, the efficient combination of DCT and motion estimation units into a single component along with the storage saving of an entire image [19] make our MPEG compatible video coder design on single chip feasible. Based on the previously defined fully pipelined parallel CORDIC architectures, we will present our VLSI implementation of the design on a dedicate single chip in this section. The chip employs a highly hierarchical and modular design. The actual design proceeds in a bottom-up manner. However, for the clarity of presentation, we use the top-down method of description to present our design.

A. Design Criteria

1) Accuracy Criterion: Of several criteria which help in determining various chip parameters to be used, perhaps the most important one is the accuracy criterion reflected in the choice of word-length. The choice of word-length will directly affect silicon area, speed and power consumption determined by the total switching activities inside the operators as well as the total effective capacitances. On the other hand, an underestimated word-length will degrade the system performance due to the increased "roundoff errors." Therefore, we carefully determined the minimum allowable system word-length that meets the accuracy criteria for cost-effective and power-saving VLSI implementations.

An input pixel represented as a fixed point finite precision number undergoes truncation at various stages of motion estimation computation, as shown in Fig. 3. This introduces noise depending on the amount of truncation. There are two basic considerations in the fixed-point analysis. One is the "roundoff error" behavior. The other is the "dynamic range." Based on the simulations, we observe that the "*2-D-DXT-II*" module (time-recursive 2-D DCT) has much higher accuracy requirement than the rest modules in our embedded video coding engine design, as shown in Fig. 3. Therefore the choice of word-length for that particular module determines the overall system accuracy. Due to the lattice structure of our design in computing those DCT coefficients, the input data undergoes the unitary transform of sinusoidal rotation in "*2-D-DXT-II*" module [37], [47]. Therefore, the overflow or dynamic range problem is not an issue. Furthermore, the orthogonal plane rotation used in that module is numerically stable so that the

TABLE II
OPTIMAL WORD-LENGTH ASSIGNMENT WITH THE CONSTRAINT OF PSNR = 40 dB

DCT channel (k)	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
B_k (N=8)	12	11	10	9	10	11	12	/	/	/	/	/	/	/	/
B_k (N=16)	13	12	12	11	11	10	10	10	10	10	11	11	12	12	13

TABLE III
THROUGHPUT AND HARDWARE COMPLEXITY OF CORDIC-HDXT-ME

Component	CORDICs	Adders	Delays (z^{-1})	Registers	Throughput
2D-DXT/IDXT	$6N$	$25N$	$6N$	$N+6N^2$	$O(N)$
Conversion	$4N$	0	0	0	$O(N)$
Pseudo Phase	$10N$	$2N$	0	0	$O(N)$
Peak Searching	0	0	$2N+2$	$2N^2$	$O(N)$
Half-pel Estimator	$2N+6$	$4N+18$	$2N+6$	$3N+N^2$	$O(N)$
Total	$22N+6$	$31N+18$	$10N+8$	$4N+9N^2$	$O(N)$

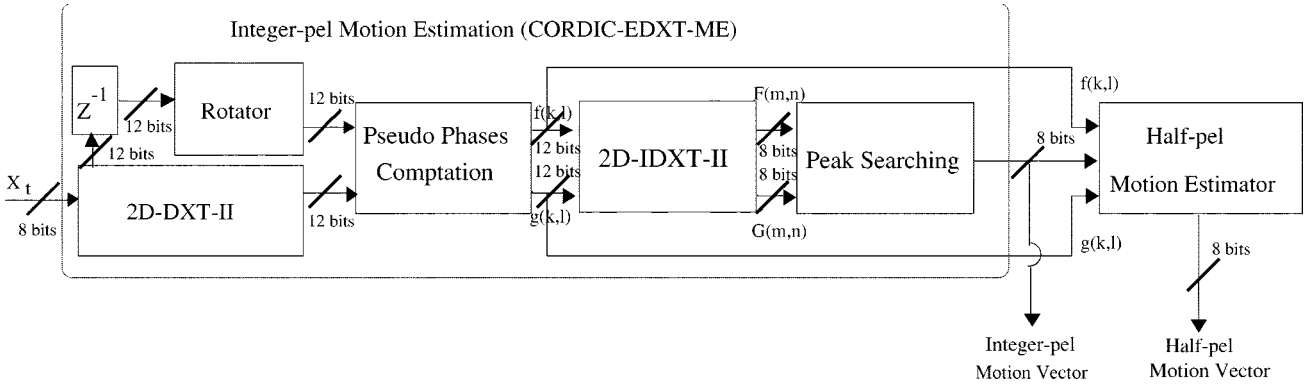

Fig. 11. Word-length assignment in CORDIC-HDXT-ME architecture.

roundoff errors will not be accumulated. Those nice properties are very useful in our fixed-point implementation.

Based on the above observations, here we determine the word-length of our design by only considering the peak signal-to-noise ratio (PSNR) and the average SNR of DCT computation due to the roundoff errors. Random data and the video test sequences such as "Miss America," "Claire," "News," and "Table Tennis" have been used as the inputs of our simulations. In order to achieve the required accuracy criterion of PSNR = 40 dB adopted by video coding standards, the optimal word-length assignment B_k for different channel k in DCT computation based on simulation results is listed in Table II. The average SNR as a function of channel number k in the fixed-point analysis is plotted in Fig. 10. Based on the experimental results, we choose 12-bit two's complement realization in our design. The word-length assignment of CORDIC-HDXT-ME architecture is outlined in Fig. 11.

Due to our CORDIC-based design, CORDIC in the different operation modes such as circular forward, circular backward, etc. [38] have been employed. In CORDIC algorithm, the number of iterations used to compute the rotation angles affects the dynamic range of design. The effects of the angle approximation error and the rounding error on the accuracy of the CORDIC computations have been analyzed in [39]. Based on that, we performed the worst case and statistical analysis of the dynamic range requirements in terms of minimum number of iterations required by different CORDICs within each module. The rounding error is computed by comparing the outputs from our float-point C model with the outputs from out fix-point implementation.

2) Speed and Area Criteria: According to the requirements of different applications, different image formats have been specified [40]. In order to make our design compatible with those MPEG video coding system, we have to satisfy the common intermediate format (CIF) and source input format (SIF) video source rate requirement. Our goal is to let the chip function at 20 MHz to achieve the international radio consultative committee (CCIR) quality. To achieve the required system throughput for different applications, careful design to optimize the speed is required. Because our proposed architecture is fully pipelined, the speed-up can be achieved by inserting latches between the consecutive modules in Fig. 12. Within each module, we can retime [41] the design by inserting latches between the successive processing units, such as CORDICs, to reduce the critical path.

The other important criterion is the chip area because it is very important to have a compact design as the fabrication cost for huge area-layout is prohibitively expensive. Some MPEG video coders available on market require multiple chips design because the silicon areas of DCT and block-based motion esti-

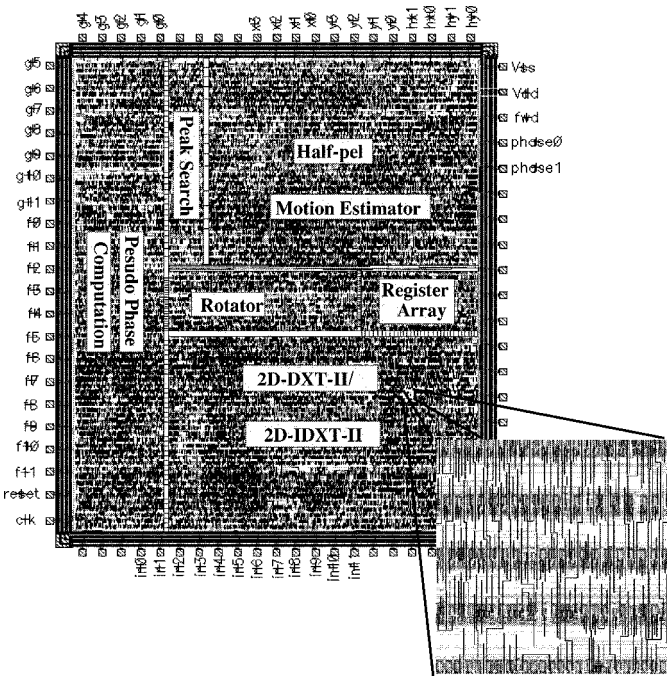

Fig. 12. Video coding engine chip. (The zoom-in picture is to show the detail layout.)

TABLE IV
VIDEO CODING ENGINE LAYOUT RESULT

Width (μm)	7036.30
Height (μm)	8652.80
Cell Area Utilization	45.82%
Cell Area(mm^2)	27.89
Design Area (mm^2)	60.88
Standard Cell Instances	38113

mation units along with the storage space for an entire image frame are too big to accommodate on the same chip [3], [19]. However, due to our DCT-based design as we have mentioned in the previous section, it is feasible to implement our MPEG compatible video coder design on a single dedicated chip based on reduced memory requirement to store the image frame [19] and the combined design of DCT and motion estimation units.

B. VLSI Implementation

Our embedded video coding engine design is quite complex. At its most detailed level, the chip consists of hundred thousand of elements as we view a system as a collection of logic gates or transistors. From a more abstract point of view, these elements can be grouped into a handful of functional components such as CORDIC processors, adders and registers. The hardware cost in terms of functional components and throughput of each stage in our design are summarized in Table III.

Overall, $20N$ CORDIC processors and $27N$ adders are required to get integer-pel motion vectors, and additional $2N+6$ CORDICs, $4N+18$ adders are required for half-pel motion estimation. The total number of operations in our design are 7168 with the block size of 16×16 and 28 672 with the block size of 32×32.

Cadence circuit design tools have been used to help us in designing such a complex system. The design begins with the conceptual idea of CORDIC-HDXT-ME architecture. It is then divided into many smaller sub-modules. We code them in Verilog [42]. A set of constraints/criteria that the final implementation must meet are applied during the synthesis procedure and *Synergy* is our synthesis tool. Also, a set of primitive components from 0.8-μm CMOS technology library are used to generate the final netlist. We select *Preview Cell Ensemble*—a "standard-cell" approach as our layout tool for our dedicated chip design. The final embedded video coding engine chip layout is shown in Fig. 12. The reason why we use 0.8-μm technology library is because that is what we had at that time. The layout results are shown in Table IV. The chip consists of 38 K gate counts. Based on timing analysis results, the chip functions at 19.73 MHz or 157.82 Mbps. If we use more advanced 0.25-μm CMOS technology, we can achieve the HDTV quality at the data rate of 663.6 Mbps with the same design.

Table V lists the comparison of previously reported implementations with our design. It indicates that our chip size is smaller than or about the same as those of block-based motion estimation design *without* DCT. Our chip naturally accommodates both DCT and motion estimation units while the others require multiple chips.

IV. CONCLUSION

Our CORDIC-based half-pel precision processing unit is developed based on the observation that half-pixel motion information is preserved in DCT coefficients. This design results in significant hardware savings and far less data flow compared to the conventional block matching methods based on interpolated images. For instance, 28 672 operations is need in our design versus 130 816 operations in the conventional half-pel block matching.

The resulted architecture has inherently massive parallel operations. Furthermore the structure is regular, modular, and has solely local connection suitable for VLSI implementation. Based on the different design criteria such as accuracy, speed and area requirements, we implement our design on a dedicated single chip to demonstrate its performance. With 0.8-μm CMOS technology, our design can operate at 20 MHz to achieve the CCIR quality (By using more advanced CMOS technology, we are able to achieve the HDTV quality with the same design). Furthermore, our implementation accommodates both DCT and motion estimation units on same chip while the conventional design may requires multiple chips. Overall, we provide a low-complexity, high throughput solution in this paper for MPEG-1, MPEG-2, and H.263 compatible video codec design.

TABLE V
COMPARISON OF OUR DESIGN WITH THOSE REPORTED MOTION ESTIMATION CIRCUITS

	Architecture	Process (μm)	#Trans	Chip Size (mm^2)	Core Size (mm^2)	Comments
[6]	BM	1.25		70	54	Bit serial without DCT
[8]	BM	1.2	52,151	64	51	without DCT
[5]	BM	1.2	27,000	86	68	without DCT
[7]	BM	1.5		98	79	without DCT
[43]	BM	0.8	540,000	125	104	Half pel without DCT
[44]	BM	0.8	405,000	648	161	Configurable design without DCT
[10]	Hie	1.2	140,000	66	51	Hierarchical BM without DCT
Our design	Trans	0.8	381,130		61	DCT+ME

† BM: Block-matching, Hie: Hierarchical BM, Trans: Transform domain approach.

REFERENCES

[1] V. Bhaskaran and K. Konstantinides, *Image and Video Compression Standards: Algorithms and Architectures*. Norwell, MA: Kluwer, 1994.

[2] K. K. Chan and C.-Y. Tsui, "Exploring the power consumption of different motion estimation architectures for video compression," in *Proc. IEEE Int. Symp. Circuits Syst.*, July 1997, pp. 1217–1220.

[3] P. Pirsch, N. Demassieux, and W. Gehrke, "VLSI architectures for video compression—A survey," *Proc. IEEE*, vol. 83, pp. 220–245, Feb. 1995.

[4] T. Xanthopoulos and A. P. Chandrakasan, "A low-power IDCT macrocell for MPEG-2 MP@ML exploiting data distribution properties for minimal activity," *IEEE J. Solid-State Circuits*, vol. 34, pp. 693–703, May 1999.

[5] O. Colavin, A. Artieri, J. F. Naviner, and R. Pacalet, "A dedicated circuit for real-time motion estimation," in *Proc. Eur. ASIC*, Feb. 1991, pp. 540–543.

[6] R. Dianysian and R. L. Baker, "Bit-serial architecture for real-time motion compensation," in *Proc. SPIE Visual Commun. Image Process.*, vol. 1001, Oct. 1988, pp. 900–907.

[7] P. Ruetz, P. Tong, D. Bailey, D. Luthi, and P. Ang, "A high-performance full-motion video compression chip set," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 2, pp. 111–122–, June 1992.

[8] K.-M. Yang, M.-T. Sun, and L. Wu, "A family of VLSI designs for the motion compensation block-matching algorithms," *IEEE Trans. Circuits Syst.*, vol. 36, pp. 1317–1325, Oct. 1989.

[9] H. Yeo and Y. H. Hu, "A novel modular systolic array architecture for full-search block matching motion estimation," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 5, pp. 407–416, Oct. 1995.

[10] R. Hervigo, J. Kowalczuk, and D. Mlynek, "A multiprocessor architecture for HDTV motion estimation system," *IEEE Trans. Consumer Electron.*, vol. 38, pp. 690–697, Aug. 1992.

[11] H. Yeo and Y. H. Hu, "A modular high-throughput architecture for logarithmic search block-matching motion estimation," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 8, pp. 299–315, June 1998.

[12] H. S. Malvar and D. H. Staelin, "The LOT: Transform coding without blocking effects," *IEEE Trans. Acoust., Speech, Signal Processing*, vol. 37, pp. 553–559, Apr. 1989.

[13] R. W. Young and N. G. Kingsbury, "Frequency-domain motion estimation using a complex lapped transform," *IEEE Trans. Image Processing*, vol. 2, pp. 2–17, Jan. 1993.

[14] J. D. Robbins and A. N. Netravali, "Recursive motion compensation: A review," in *Image Sequence Processing and Dynamic Scene Analysis*, T. S. Huang, Ed. Berlin, Germany: Springer-Verlag, 1983, pp. 76–103.

[15] R. C. Kim and S. U. Lee, "A VLSI architecture for a pel recursive motion estimation algorithm," *IEEE Trans. Circuits Syst.*, vol. 36, pp. 1291–1300, Oct. 1989.

[16] A. Singh, *Optic Flow Computation—A Unified Perspective*. Piscataway, NJ: IEEE Computer Society Press, 1991.

[17] S.-F. Chang and D. G. Messerschmitt, "Manipulation and compositing of MC-DCT compressed video," *IEEE J. Select. Areas Commun.*, vol. 13, pp. 1–11, Jan. 1995.

[18] A. Kojima, N. Sakurai, and J. Kishigami, "Motion detection using 3D-FFT spectrum," in *Proc. IEEE Int. Conf. Acoustics, Speech, Signal Process.*, Apr. 1993, pp. V213–V216.

[19] R. Kleihorst and F. Cabrera, "Implementation of DCT-domain motion estimation and compensation," in *IEEE Workshop Signal Process. Syst.*, 1998, pp. 53–62.

[20] B. Girod, "Why B-pictures work: A theory of multi-hypothesis motion-compensated prediction," in *Proc. 1998 IEEE Int. Conf. Image Process.*, Oct. 1998, pp. 213–217.

[21] S.-L. Iu, "Comparison of motion compensation using different degrees of sub-pixel accuracy for interfield/interframe hybrid coding of HDTV image sequences," in *Proc. IEEE Int. Conf. Acoustics, Speech, Signal Process.*, vol. 3, 1992, pp. 465–468.

[22] B. Girod, *Motion Compensation: Visual Aspects, Accuracy, and Fundamental Limits*. Norwell, MA: Kluwer, 1993.

[23] J. K. Aggarwal and N. Nandhakumar, "On the computation of motion from sequences of images—A review," *Proc. IEEE*, vol. 76, pp. 917–935, Aug. 1988.

[24] I. J. Cox, J. B. Kruskal, and D. A. Wallach, "Predicting and estimating the accuracy of a subpixel registration algorithm," *IEEE Trans. Pattern Anal. Machine Intell.*, vol. 12, pp. 721–734, Aug. 1990.

[25] H. G. Musmann, P. Pirsch, and H.-J. Grallert, "Advances in picture coding," *Proc. IEEE*, vol. 73, pp. 523–548, Apr. 1985.

[26] G. de Haan and W. A. C. Biezen, "Sub-pixel motion estimation with 3-D recursive search block-matching," *Signal Process. Image Commun.*, vol. 6, pp. 229–239, June 1994.

[27] B. Girod, "Motion-compensating prediction with fractional-pel accuracy," *IEEE Trans. Commun.*, vol. 41, pp. 604–612, Apr. 1993.

[28] G. Madec, "Half pixel accuracy in block matching," in *Picture Coding Symp.*, Cambridge, MA, Mar. 1990.

[29] A. N. Netravali and J. D. Robbins, "Motion compensated television coding—Part 1," *Bell Syst. Tech. J.*, vol. 58, pp. 631–670, Mar. 1979.

[30] A. Nosratinia and M. T. Orchard, "Discrete formulation of pel-recursive motion compensation with recursive least squares updates," in *Proc. IEEE Int. Conf. Acoustics, Speech, Signal Process.*, vol. 5, 1993, pp. 229–232.

[31] R. W. Schafer and L. R. Rabiner, "A digital signal processing approach to interpolation," *Proc. IEEE*, pp. 692–702, June 1973.

[32] K. Ishihara, S. Masuda, S. Hattori, H. Nishikawa, Y. Ajioka, T. Yamada, H. Amishiro, S. Uramoto, M. Yoshimoto, and T. Sumi, "A half-pel precision MPEG2 motion-estimation processor with concurrent three-vector search," *IEEE J. Solid-State Circuits*, vol. 30, pp. 1502–1509, Dec. 1995.

[33] U.-V. Koc and K. J. R. Liu, "Discrete-cosine/sine-transform based motion estimation," in *Proc. IEEE Int. Conf. Image Process.*, vol. 3, Austin, TX, Nov. 1994, pp. 771–775.

[34] ——, "DCT-based subpixel motion estimation," in *Proc. IEEE Int. Conf. Acoustics, Speech, Signal Process.*, 1996, pp. 1930–1933.

[35] K. R. Rao and P. Yip, *Discrete Cosine Transform: Algorithms, Advantages, and Applications*. New York: Academic, 1990.

[36] Z. Wang, "Fast algorithms for the discrete W transform and for the discrete Fourier transform," *IEEE Trans. Acoust., Speech, Signal Processing*, vol. ASSP-32, pp. 803–816, Aug. 1984.

[37] J. Chen and K. J. R. Liu, "A complete pipelined parallel CORDIC architecture for motion estimation," *IEEE Trans. Circuits Syst. II*, vol. 45, pp. 653–660, May 1998.

[38] Y. H. Hu, "CORDIC-based VLSI architectures for digital signal processing," *IEEE Signal Processing Mag.*, pp. 16–35, July 1992.

[39] ——, "The quantization effects of the CORDIC algorithms," *IEEE Trans. Signal Processing*, vol. 40, pp. 834–844, Apr. 1992.

[40] C. R. MPEG-2, "Generic coding of moving pictures and associated audio,", Geneve, Switzerland, ISO/IEC 13 818 H.262, 1994.

[41] L. A. Glasser and D. W. Dobberpuhl, *The Design and Analysis of VLSI Circuits*. Reading, MA: Addison-Wesley, 1985.

[42] D. J. Smith, *HDL Chip Design—A Practical Guide for Designing, Synthesizing and Simulating ASICs and FPGAs using VHDL or Verilog*. Madison, AL: Doone, 1997.

[43] D. J. Uramoto *et al.*, "A half-pel precision motion estimation processor for NTSC-resolution video," in *Proc. IEEE Custom Integrated Circuits Conf.*, June 1993, pp. 11.2.1–11.2.4.
[44] Y. Tokuno *et al.*, "A motion video compression LSI with distributed arithmetic architecture," in *Proc. IEEE Custom Integrated Circuits Conf.*, June 1993, pp. 11.3.1–11.3.4.
[45] U.-V. Koc and K. J. R. Liu, "Discrete-cosine/sine-transform based motion estimation," *IEEE Trans. Image Processing*, July 1998.
[46] ——, "DCT-based subpixel motion estimation," *IEEE Trans. Circuits Syst. Video Technol.*, Aug. 1998.
[47] J. Chen and K. J. R. Liu, "A complete pipelined parallel CORDIC architecture for motion estimation," in *Proc. Int. Symp. Circuits Syst.*, Hong-Kong, July 1997.

Jie Chen (S'95–M'97) received the B.S. degree in electrical engineering and the M.S. degree in physics from Fudan University, Shanghai, China, in 1987 and in 1990, respectively, and the M.S. and Ph.D. degrees in electrical engineering from the University of Maryland, College Park, in 1992 and 1998, respectively.

From 1992 to 1995, he was with Hughes Network System Research Group, Germantown, MD, and participated in TDMA and multimedia over wireless local-loop system design and implementation. Later, he joined the ITU-T working group developing the third-generation CDMA standard. From 1998 to 2000, he was with Bell Laboratories, Lucent Technologies, Murray Hill, NJ, where he was involved in research and development of optical network integrated circuits. He is currently a Chief Hardware Architecture and Principle System Engineer of a start-up company, working on digital audio broadcast and wireless mobile data network. His research interests include multimedia signal processing, multimedia communications, high-speed data networking design, and management. He has published over 15 papers and holds three U.S. patents. He is the author of the book *Design of Digital Video Coding Systems: A Complete Compressed Domain Approach* (New York: Marcel Dekker, 2001.

Dr. Chen serves as an Editor for *EURASIP Journal of Applied Signal Processing* and was a Guest Editor of a Special Issue on "Multimedia Over IP" of the IEEE TRANSACTIONS ON MULTIMEDIA. He was the Special Chair of "Multimedia Over Networks" and held a tutorial session of "Understanding Emerging and Future Communication Technologies" in the International Symposium of Circuits and System 2000 (ISCAS'00) and ISCAS 2001, respectively. He also serves as a Technical Member of both multimedia and video technologies committees for the IEEE Circuits and Systems Society.

K. J. Ray Liu (S'86–M'86–SM'93) received the B.S. degree from the National Taiwan University, Taipei, Taiwan, R.O.C., in 1983, and the Ph.D. degree from the University of California, Los Angeles, in 1990, both in electrical engineering.

Since 1990, he has been with the Electrical and Computer Engineering Department and Institute for Systems Research, University of Maryland, College Park, where he is a Professor. During his sabbatical leave in 1996 and 1997, he was Visiting Associate Professor at Stanford University, Stanford, CA. His research interests span broad aspects of signal processing, image/video processing, and communications in which he has published over 200 papers, of which over 60 are in archival journals.

Dr. Liu has received numerous awards including the 1994 National Science Foundation Young Investigator Award, the IEEE Signal Processing Society's 1993 Senior Award (Best Paper Award), the IEEE Vehicular Technology Conference (VTC) 1999 Best Paper Award, Amsterdam, the George Corcoran Award in 1994 for outstanding contributions to electrical engineering education, and the 1995–1996 Outstanding Systems Engineering Faculty Award, in recognition of outstanding contributions in interdisciplinary research, both from the University of Maryland, and many others. He is Editor-in-Chief of *EURASIP Journal on Applied Signal Processing*, and has been an Associate Editor of IEEE TRANSACTIONS ON SIGNAL PROCESSING, a Guest Editor of special issues on Multimedia Signal Processing of PROCEEDINGS OF THE IEEE, a Guest Editor of a special issue on Signal Processing for Wireless Communications of IEEE JOURNAL OF SELECTED AREAS IN COMMUNICATIONS, a Guest Editor of special issue on Multimedia Communications over Networks of *IEEE Signal Processing Magazine*, a Guest Editor of special issue on Multimedia over IP of IEEE TRANSACTIONS ON MULTIMEDIA, and an Editor of *Journal of VLSI Signal Processing Systems*. He currently serves as the Chair of Multimedia Signal Processing Technical Committee of IEEE Signal Processing Society and the Series Editor of Marcel Dekker series on Signal Processing and Communications.